

Figure 2

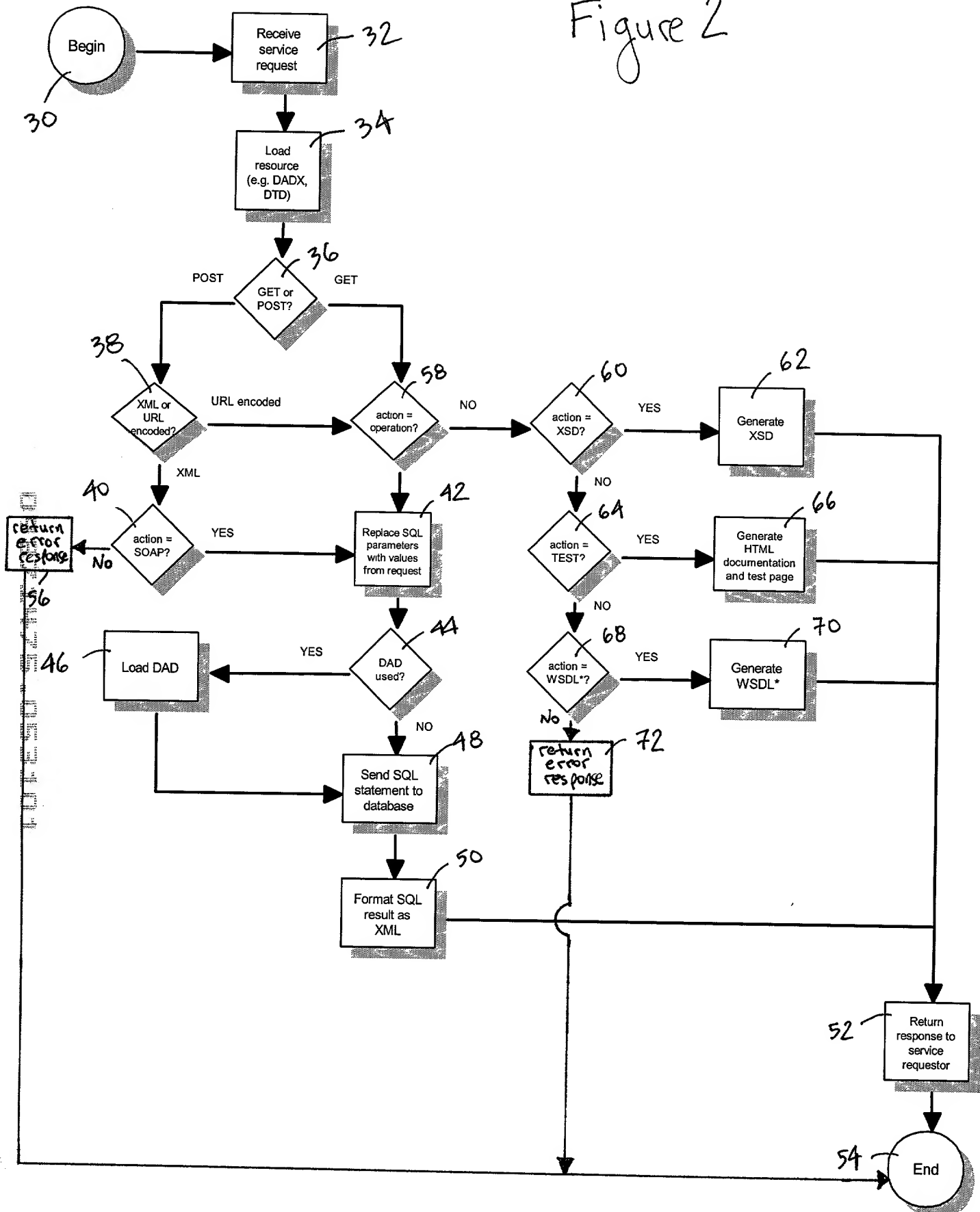


Figure 3

```

<?xml version="1.0"?>
<DADX xmlns="urn:ibm.com:dadx"
  xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">
  <documentation>mycompany part orders service.</documentation>

70  <operation name="findAll">
    <documentation>Returns all the orders.</documentation>
    <retrieveXML>
      <DAD_ref>getstart_xcollection.dad</DAD_ref>
76    <SQL_override>
      select o.order_key, customer_name, customer_email,
        p.part_key, color, quantity, price, tax, ship_id, date, mode
      from order_tab o, part_tab p,
78      table(select substr(char(timestamp(generate_unique())),16) as ship_id,
        date, mode, part_key from ship_tab) s
84    where p.order_key = o.order_key and s.part_key = p.part_key
      order by order_key, part_key, ship_id
    </SQL_override>
    </retrieveXML>
  </operation>

72  <operation name="findByColor">
    <documentation>Returns all the orders that have the specified color.</documentation>
    <retrieveXML>
      <DAD_ref>getstart_xcollection.dad</DAD_ref>
76    <SQL_override>
      select o.order_key, customer_name, customer_email,
        p.part_key, color, quantity, price, tax, ship_id, date, mode
      from order_tab o, part_tab p,
80    table(select substr(char(timestamp(generate_unique())),16) as ship_id,
        date, mode, part_key from ship_tab) s
86    where p.order_key = o.order_key and s.part_key = p.part_key
      and color = :color
      order by order_key, part_key, ship_id
    </SQL_override>
    <parameters>
      <parameter name="color" type="xsd:string"/>
    </parameters>
    </retrieveXML>
  </operation>

74  <operation name="findByMinPrice">
    <documentation>Returns all the orders that have a price greater than or equal to the
      specified minimum price.</documentation>
    <retrieveXML>
      <DAD_ref>getstart_xcollection.dad</DAD_ref>
76    <SQL_override>
      select o.order_key, customer_name, customer_email,
        p.part_key, color, quantity, price, tax, ship_id, date, mode
      from order_tab o, part_tab p,
82    table(select substr(char(timestamp(generate_unique())),16) as ship_id,
        date, mode, part_key from ship_tab) s
88    where p.order_key = o.order_key and s.part_key = p.part_key
      and p.price >= :minprice
      order by order_key, part_key, ship_id
    </SQL_override>
    <parameters>
      <parameter name="minprice" type="xsd:decimal"/>
    </parameters>
    </retrieveXML>
  </operation>
</DADX>

```

FIG. 4

```

<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/2000/10/XMLSchema"
  targetNamespace="urn:ibm.com:dxx:dadx"
  xmlns:dadx="urn:ibm.com:dxx:dadx">

  <annotation>
    <documentation>A DADX document defines a Web Service implemented by operations that use
      the DB2 XML Extender.
    </documentation>
  </annotation>

  <element name="DADX">

    <complexType>
      <sequence>
        <element ref="dadx:documentation" minOccurs="0"/>
        <element ref="dadx:implements" minOccurs="0"/>
        <element ref="dadx:operation" maxOccurs="unbounded"/>
      </sequence>
    </complexType>

    <unique name="operationNames">
      <selector xpath="dadx:operation"/>
      <field xpath="@name"/>
    </unique>

  </element>

  <element name="documentation">
    <complexType mixed="true">
      <choice minOccurs="0" maxOccurs="unbounded">
        <any minOccurs="0" maxOccurs="unbounded"/>
      </choices>
      <anyAttribute/>
    </complexType>
  </element>

  <element name="implements">
    <complexType>
      <attribute name="namespace" type="uriReference" use="required"/>
      <attribute name="location" type="uriReference" use="required"/>
    </complexType>
  </element>

  <element name="operation">
    <complexType>
      <sequence>
        <element ref="dadx:documentation" minOccurs="0"/>
        <choice>
          <element ref="dadx:retrieveXML"/>
          <element ref="dadx:storeXML"/>
          <element ref="dadx:query"/>
          <element ref="dadx:update"/>
          <element ref="dadx:call"/>
        </choice>
      </sequence>
      <attribute name="name" type="NCName" use="required"/>
    </complexType>
  </element>

```

104
106
108

FIG. 4 (cont.)

<element name="retrieveXML">

<complexType>

<sequence>

<choice>

<element ref="dadx:DAD_ref"/>

<element ref="dadx:collection_name"/>

</choice>

<choice>

<element name="no_override">

<complexType/>

</element>

112 <element name="SQL_override" type="string"/>

<element name="XML_override" type="string"/>

</choice>

110 <element ref="dadx:parameter" minOccurs="0" maxOccurs="unbounded"/>

</sequence>

</complexType>

<unique name="parameterNames">

<selector xpath="dadx:parameter"/>

<field xpath="@name"/>

</unique>

</element>

<element name="storeXML">

<complexType>

<choice>

<element ref="dadx:DAD_ref"/>

<element ref="dadx:collection_name"/>

</choice>

</complexType>

</element>

<element name="query">

<complexType>

<sequence>

114 <element name="SQL_query" type="string"/>

<element ref="dadx:XML_result" minOccurs="0" maxOccurs="unbounded"/>

110 <element ref="dadx:parameter" minOccurs="0" maxOccurs="unbounded"/>

</sequence>

</complexType>

<unique name="XML_resultNames">

<selector xpath="dadx:XML_result"/>

<field xpath="@name"/>

</unique>

<unique name="parameterNames">

<selector xpath="dadx:parameter"/>

<field xpath="@name"/>

</unique>

</element>

<element name="update">

<complexType>

<sequence>

116 <element name="SQL_update" type="string"/>

110 <element ref="dadx:parameter" minOccurs="0" maxOccurs="unbounded"/>

</sequence>

</complexType>

<unique name="parameterNames">

<selector xpath="dadx:parameter"/>

<field xpath="@name"/>

</unique>

FIG. 4 (cont.)

```

</element>

<element name="call">
  <complexType>
    <sequence>
      118 <element name="SQL_call" type="string"/>
      110 <element ref="dadx:parameter" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </complexType>

  <unique name="parameterNames">
    <selector xpath="dadx:parameter"/>
    <field xpath="@name"/>
  </unique>
</element>

<element name="no_override">
  <complexType/>
</element>

<element name="DAD_ref" type="string"/>

<element name="collection_name" type="string"/>

<element name="parameter">
  <complexType>
    <attribute name="name" type="NCName" use="required"/>
    <attribute name="element" type="QName"/>
    <attribute name="type" type="QName"/>
    <attribute name="kind" type="dadx:parameterKindType" default="in"/>
  </complexType>
</element>

<simpleType name="parameterKindType">
  <restriction base="string">
    <enumeration value="in"/>
    <enumeration value="out"/>
    <enumeration value="in/out"/>
  </restriction>
</simpleType>

<element name="XML_result">
  <complexType>
    <attribute name="name" type="NCName" use="required"/>
    <attribute name="element" type="QName" use="required"/>
  </complexType>
</element>

</schema>

```